Model Results for GDP ~ Energy

Grant Esparza

4/26/2018

```r
## Libraries used
library(lme4)
library(sjPlot)
library(dplyr)
library(ggplot2)
```

```r
#Read data
states.energy <- read.csv("~/Energy_Analysis/data/usa_states_energy.csv")
usa.energy <- read.csv("~/Energy_Analysis/data/usa_energy.csv")

## Clean it up a little
states.energy <- states.energy %>% select(c(year, State, CLTCB, FFTCB, GETCB, HYTCB,
                                           MGTCB, NGTCB, SOTCB, WYTCB, GDP,
                                           GDP.total))
names(states.energy)[11] <- "GDP.state"
states.energy$GDP.total.growth <- ifelse(usa.energy[38:56, 22] > 1.03,
                                          1,0)

names(states.energy)
```

```
##  [1] "year"             "State"            "CLTCB"
##  [4] "FFTCB"            "GETCB"            "HYTCB"
##  [7] "MGTCB"            "NGTCB"            "SOTCB"
## [10] "WYTCB"            "GDP.state"        "GDP.total"
## [13] "GDP.total.growth"
```

Some plotting showed that in order to keep these variables on the same scale as GDP, I needed to use a log transformation. Since all variables are transformed the analysis should be valid.

```r
## Perform a log transformation on all variables
for (i in 1:8){
    states.energy[, i+2] <- log(states.energy[,i+2] + 1)
}
states.energy$GDP.total <- log(states.energy$GDP.total)
states.energy$GDP.state <- log(states.energy$GDP.state)
```

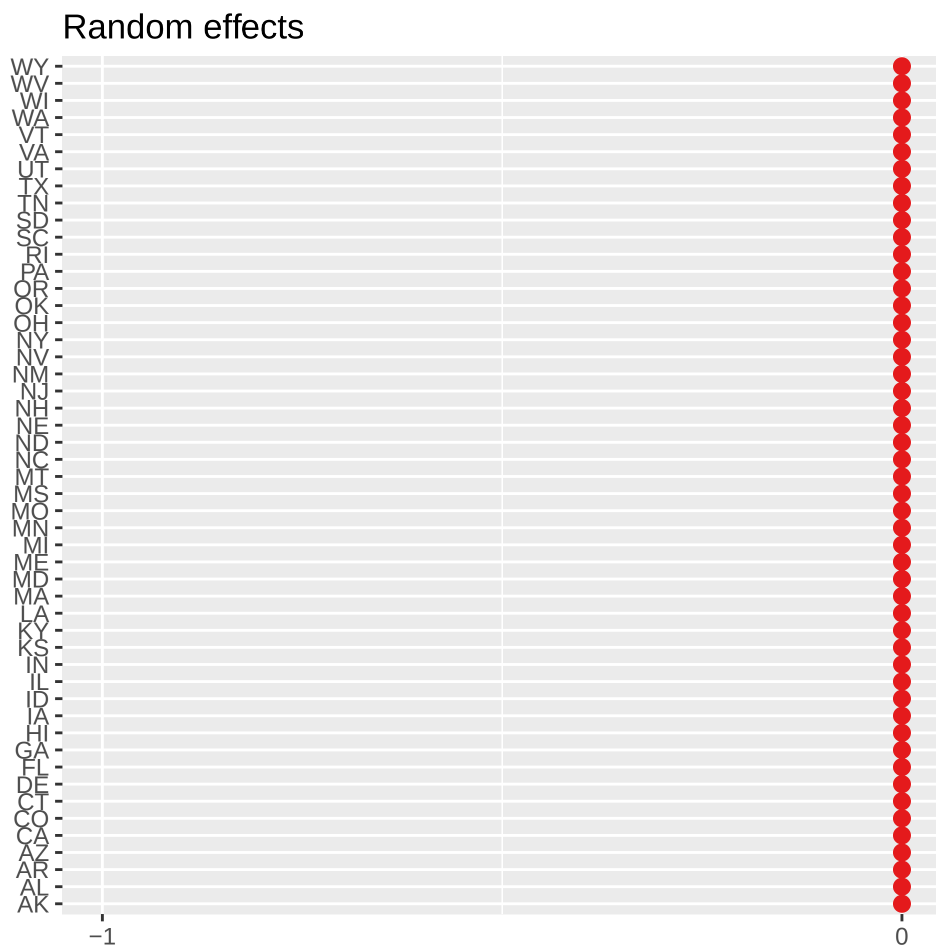
**After constant tweaking I'm pretty much in the same place in regards to the random intercept model. I talked to Edward about my model a little bit and he was also at a loss as to why the variance for the random intercept would be equal to zero. This happens when including the intercept (*see no '-1'*).

However when removing the intercept, we get meaningful results that seem to reflect what I've been able to generate in plots(*see the density plots at the end.*) Edward says this could be a failing of frequentist statistics, and that for now we should run with *log(GDP.state) ~ -1 + log(HYTCB) + (1|State)*. We should ask Robin about this tomorrow.**

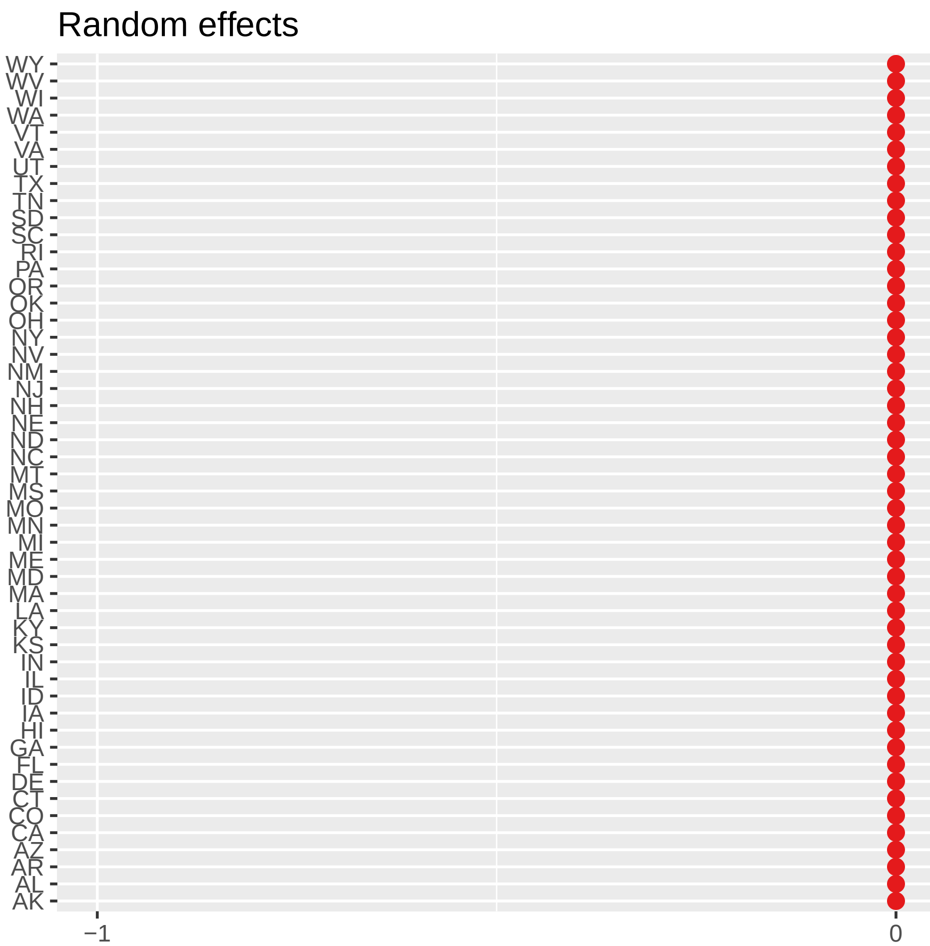
```r
## Partial Pooling
contin.model.intercept <- lmer(GDP.state ~ HYTCB + (1 | State),
                               data=states.energy)
```

```
contin.model.no_intercept <- lmer(GDP.state ~ -1 + HYTCB +
                                  (1 | State), data=states.energy)


## Make sjPlot for random effects
plot_model(contin.model.intercept, sort.est="(Intercept)", type="re",
           y.offset=.4)
```




```
plot_model(contin.model.intercept, sort.est="(Intercept)", type="re",
           y.offset=.4)
```




```r
summary(contin.model.intercept)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: GDP.state ~ HYTCB + (1 | State)
##    Data: states.energy
##
## REML criterion at convergence: 2769.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.06713 -0.79313  0.04727  0.74016  2.52451
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  State    (Intercept) 0.00     0.000
##  Residual             1.07     1.034
## Number of obs: 950, groups:  State, 50
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 11.924347   0.121918   97.81
## HYTCB        0.009225   0.012806    0.72
##
## Correlation of Fixed Effects:
```

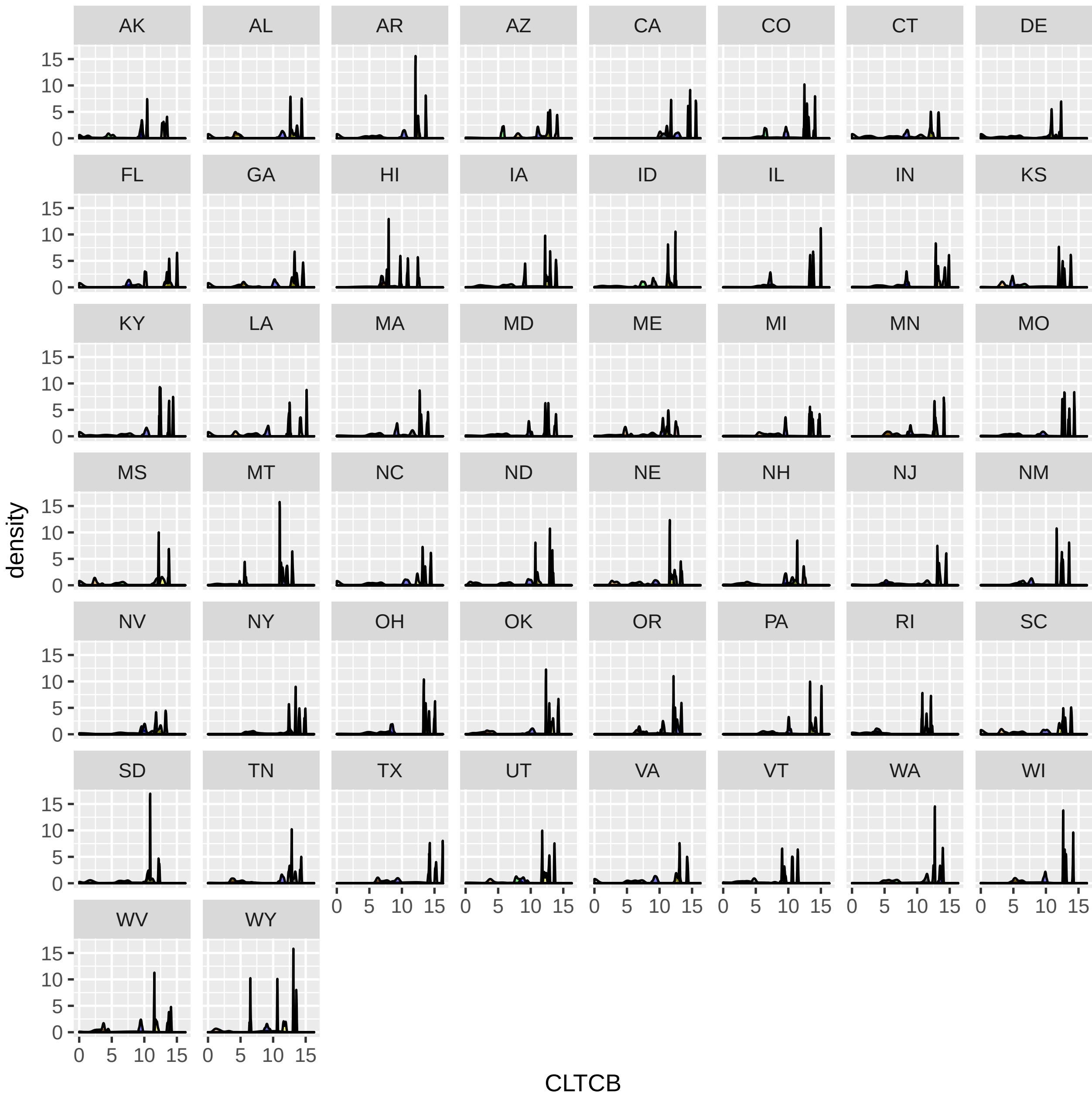
```
##       (Intr)
## HYTCB -0.961
```

```
summary(contin.model.no_intercept)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: GDP.state ~ -1 + HYTCB + (1 | State)
##    Data: states.energy
##
## REML criterion at convergence: 3125.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.02981 -0.71466  0.04706  0.69155  2.93594
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  State    (Intercept) 13.27    3.642
##  Residual              1.18    1.086
## Number of obs: 950, groups:  State, 50
##
## Fixed effects:
##       Estimate Std. Error t value
## HYTCB  1.05142    0.05049   20.82
```

As you can see in the plots below, HYTCB seems to be a prevelant contributor compared to other states. This reflects what the random intercept model shows us in the previous sjPlot. Another apparent feature of these graphs is the overwhelming domination of nonrewnewable sources. . . but we already could have guessed that coming in.

```
ggplot(states.energy) +
  geom_density(aes(x=CLTCB), fill="black", alpha=.5) +
  geom_density(aes(x=FFTCB), fill="brown", alpha=.5) +
  geom_density(aes(x=GETCB), fill="green", alpha=.5) +
  geom_density(aes(x=HYTCB), fill="blue", alpha=.5) +
  geom_density(aes(x=MGTCB), fill="gray", alpha=.5) +
  geom_density(aes(x=NGTCB), fill="yellow", alpha=.5) +
  geom_density(aes(x=SOTCB), fill="orange", alpha=.5) +
  geom_density(aes(x=WYTCB), fill="light blue", alpha=.5) +
  facet_wrap(~State)
```

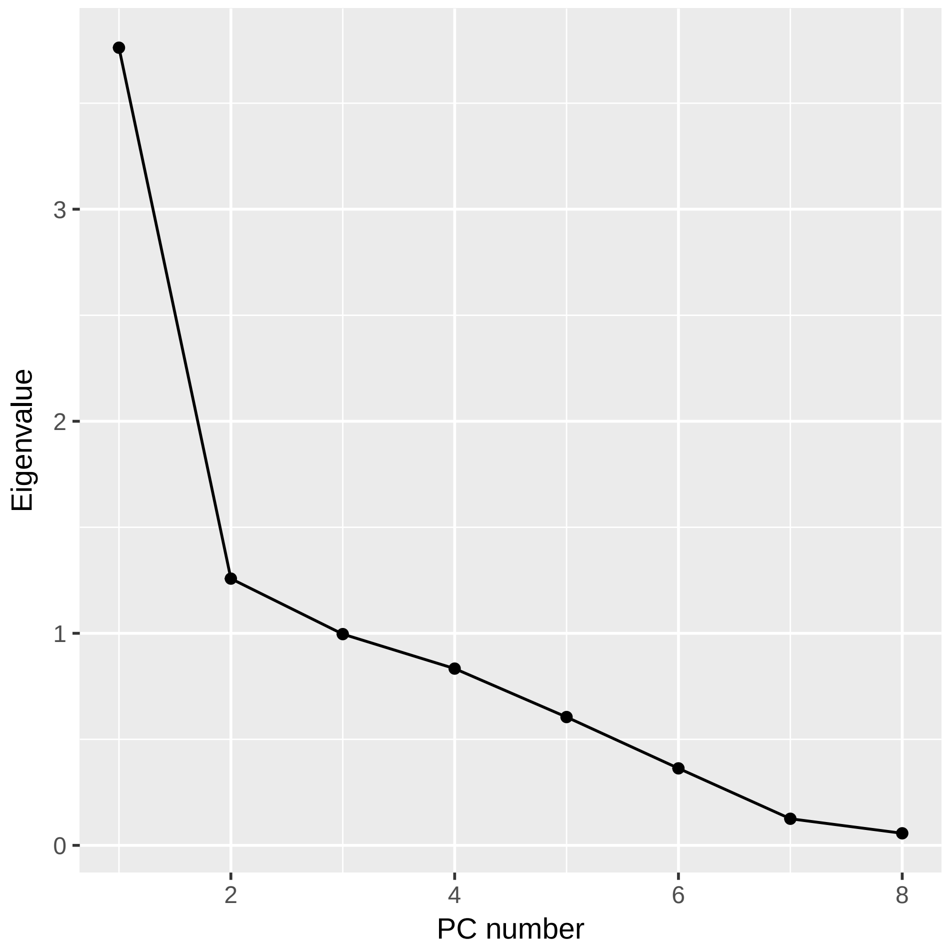


I also went ahead and ran a factor analysis to see if nonrewnewable and renewable energy ended up in distinct groups or "factors". At least in regards to their total consumption the plot below shows some pretty distinct grouping.

```
# Factor Analysis
library(corrplot)
library(psych)

states.dat <- states.energy %>% select(c(CLTCB, FFTCB, GETCB, HYTCB, MGTCB,
                                          NGTCB, SOTCB, WYTCB))

extract_pca <- princomp(states.dat, cor=TRUE)
var_pc <- (extract_pca$sdev)^2
qplot(x=1:length(var_pc), y=var_pc, geom=c("point", "line")) +
    xlab("PC number") + ylab("Eigenvalue")
```

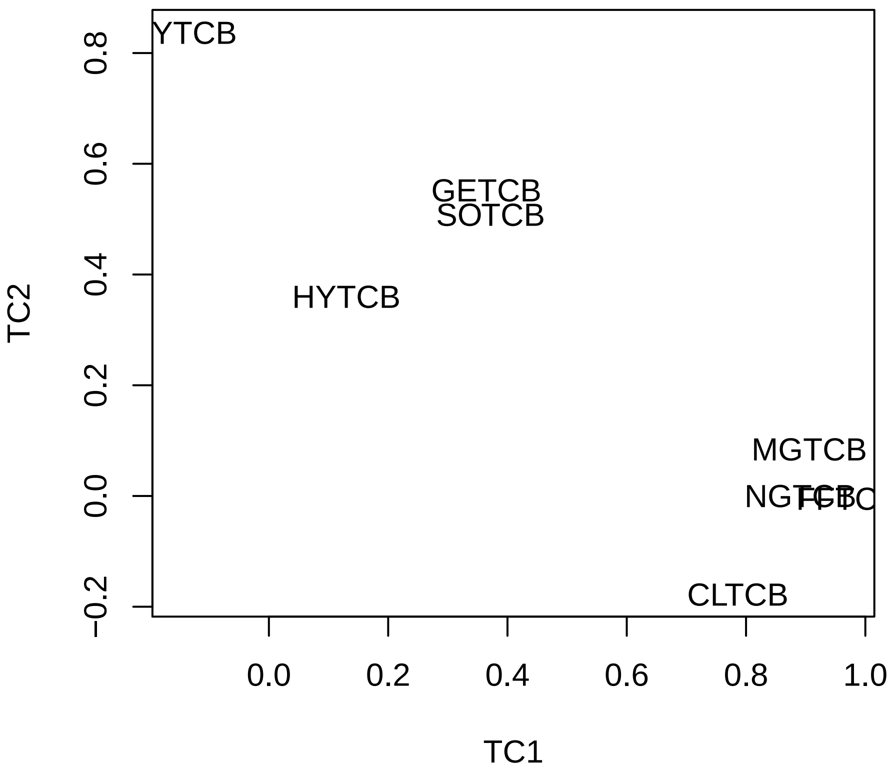



```r
pc.model.oblimin <- principal(states.dat, nfactors=2, rotate="oblimin",
                               scores=TRUE)
print(pc.model.oblimin)
```

```
## Principal Components Analysis
## Call: principal(r = states.dat, nfactors = 2, rotate = "oblimin", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##         TC1   TC2   h2   u2 com
## CLTCB  0.79 -0.18 0.59 0.41 1.1
## FFTCB  0.97  0.00 0.94 0.06 1.0
## GETCB  0.36  0.55 0.52 0.48 1.7
## HYTCB  0.13  0.36 0.17 0.83 1.3
## MGTCB  0.91  0.08 0.86 0.14 1.0
## NGTCB  0.89  0.00 0.79 0.21 1.0
## SOTCB  0.37  0.51 0.47 0.53 1.8
## WYTCB -0.15  0.84 0.67 0.33 1.1
##
##                        TC1  TC2
## SS loadings           3.54 1.48
## Proportion Var        0.44 0.19
## Cumulative Var        0.44 0.63
## Proportion Explained  0.70 0.30
```

```
## Cumulative Proportion 0.70 1.00
##
##  With component correlations of
##     TC1 TC2
## TC1 1.0 0.2
## TC2 0.2 1.0
##
## Mean item complexity =  1.3
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.11
##  with the empirical chi square  630.33  with prob <  2.6e-126
##
## Fit based upon off diagonal values = 0.93
```

```r
load <- pc.model.oblimin$loadings
plot(load, type="n")
text(load, labels=rownames(load))
```



**All in all, if we can run with the second variation of the random intercept model we can come up with some decent results. If Robin says we can't use it I still have the tables from the for loops running lm on each energy source but I reallllly don't like how those turned out.